MrBayes version 3.2 Manual

Fredrik Ronquist, John Huelsenbeck and Maxim Teslenko

July 27, 2011

1 Introduction

MrBayes 3 is a program for the Bayesian inference of phylogeny. The program has a command-line interface and should run on a variety of computer platforms, including clusters of Macintosh and UNIX computers. Note that the computer should be reasonably fast and should have a lot of RAM memory (depending on the size of the data matrix, the program may require hundreds of megabytes of memory). The program is optimized for speed and not for minimizing memory requirements. This manual explains how to use the program. After this section, which introduces the program, we will first walk you through a simple analysis (section 2 of the manual), which will get you started, and a more complex analysis that uses more of the program's capabilities (section 3). We then briefly describe the models implemented in the program (section 4), answer some frequently asked questions (section 5), and discuss the differences between versions 2 and 3 of the program (section 6). Finally, we give more detailed instructions on how to compile the program and how to run the parallel versions of it (section 7). Section 7 also contains brief information for developers interested in tweaking MrBayes code or contributing to the MrBayes project. The manual ends with a series of diagrams giving a graphical overview of all the models and proposal mechanisms implemented in the program (Appendix). For more detailed information about commands and options in MrBayes, see the command reference that can either be downloaded from the program web site or generated from the program itself (see section 1.4 Getting Help below). All the information in the command reference is also available on-line when using the program. The manual assumes that you are familiar with the basic concepts of Bayesian phylogenetics. If you are new to the subject, we recommend the recent reviews by Holder and Lewis (2003), Lewis (2001) and Huelsenbeck et al. (2001, 2002). It is also worthwhile to study the early papers introducing Bayesian phylogenetic methods (Li 1996; Mau, 1996; Rannala and Yang, 1996; Mau and Newton, 1997; Rannala and Yang, 1997; Larget and Simon, 1999; Mau, Newton and Larget, 1999; Newton, Mau and Larget, 1999). The basic MCMC techniques are described in Metropolis et al. (1953) and Hastings (1970). The Metropolis-coupled MCMC used by MrBayes was introduced by Geyer (1991).

1.1 Conventions Used in this Manual

Throughout the document, we use `typewriter font` for things you see on screen or in a data file, and **bold font** for things you should type in. Alternative commands you could have typed in, but should not type in to follow the tutorial, are also given in `typewriter font`.

1.2 Acquiring and Installing MrBayes

MrBayes 3 is distributed without charge by download from the MrBayes web site, http://mrbayes.net. If someone has given you a copy of MrBayes 3, we strongly suggest that you download the most recent version from this site. The site also gives informationabout the MrBayes users email list and describes how you can report bugs or contribute to the project.

MrBayes 3 is a plain-vanilla program that uses a command line interface and therefore behaves virtually the same on all platforms - Macintosh, Windows and Unix. There is a separate download package for each platform. The Macintosh package contains two versions of the program: the standard serial version, named MrBayes3.1 (program icon one copy of Reverend Bayes's portrait), and a version for running the program in parallel on clusters of Macintoshes, named MrBayes3.1p (program icon four portraits of Reverend Bayes). For more information on the parallel Macintosh version of MrBayes, which requires the installation of POOCH, see section 7 of this user manual. The Windows package only contains the serial version of the program and is ready to run after unzipping, just like the Macintosh serial version.

If you decide to run the program under Unix/Linux, then you need to compile the program from the source code. In the latter case, simply unpack the file `mrbayes3.2_src.tar.gz` by typing `gunzip MrBayes_3.2_src.tar.gz` and then `tar -xf MrBayes_3.2_src.tar` . The `gunzip` command unzips the compressed file and the `tar -xf` command extracts all of the files from the `.tar` archive that resulted from the unzip operation (note that the `.gz` suffix is dropped in the unzip operation). You then need to compile the program. We have included a "Makefile" that contains compiler instructions producing the serial version of the program. You simply type `make` to compile the program according to these instructions. A typical compile session would look like this:

```
ronquistg5:mrbayes>ls
mrbayes3.2\_src.tar.gz
ronquistg5:mrbayes>gunzip MrBayes\_3.2\_src.tar.gz
ronquistg5:mrbayes>ls
mrbayes3.2\_src.tar
ronquistg5:mrbayes>tar -xf MrBayes\_3.2\_src.tar
ronquistg5:mrbayes>make
gcc -DUNIX\_VERSION -O3 -Wall -Wno-uninitialized -c -o mb.o mb.c
gcc -DUNIX\_VERSION -O3 -Wall -Wno-uninitialized -c -o mcmc.o mcmc.c
gcc -DUNIX\_VERSION -O3 -Wall -Wno-uninitialized -c -o bayes.o bayes.c
gcc -DUNIX\_VERSION -O3 -Wall -Wno-uninitialized -c -o command.o command.c
gcc -DUNIX\_VERSION -O3 -Wall -Wno-uninitialized -c -o mbmath.o mbmath.c
gcc -DUNIX\_VERSION -O3 -Wall -Wno-uninitialized -c -o model.o model.c
gcc -DUNIX\_VERSION -O3 -Wall -Wno-uninitialized -c -o plot.o plot.c
gcc -DUNIX\_VERSION -O3 -Wall -Wno-uninitialized -c -o sump.o sump.c
gcc -DUNIX\_VERSION -O3 -Wall -Wno-uninitialized -c -o sumt.o sumt.c
gcc -DUNIX\_VERSION -O3 -Wall -Wno-uninitialized -lm mb.o bayes.o command.o
mbmath.o mcmc.o model.o plot.o sump.o sumt.o -o mb
ronquistg5:mrbayes>
```

The compilation usually stops for several minutes at the mcmc.c file; this is perfectly normal. This is the largest source file and optimization of the code takes quite a while.

We assume as the default C compiler gcc, which is installed on most systems. If you do not have gcc installed on your machine, or you want to produce the MPI version or some other special version of the program, you have to change the compiler information in the Makefile as described in section 7 of this manual. The executable serial version of the program is called "mb". To execute the program, simply type `./mb` in the directory where you compiled the program. The `./` prefix is needed to tell Unix that you want to run the local copy of mb in your working directory. If you run MrBayes often, you will probably want to add the program to your "path"; refer to your Unix manual or your local Unix expert for more information on this.

All three packages of MrBayes come with example data files. These are intended to show various types of analyses you can perform with the program, and you can use them as templates for your own analyses. Two of the files, primates.nex and cynmix.nex, will be used in the tutorial sections of this manual (sections 2 and 3).

1.3 Getting Started

Start MrBayes by double-clicking the application icon (or typing `./mb`) and you will see the information below:

```
                    MrBayes v3.2-cvs

                (Bayesian Analysis of Phylogeny)

            Distributed under the GNU General Public License


          Type "help" or "help <command>" for information
                on the commands that are available.

              Type "about" for authorship and general
                  information about the program.


MrBayes >
```

Note the `MrBayes >` prompt at the bottom, which tells you that MrBayes is ready for your commands.

1.4 Changing the Size of the MrBayes Window

Some MrBayes commands will output a lot of information and write fairly long lines, so you may want to change the size of the MrBayes window to make it easier to read the output. On Macintosh and Unix machines, you should be able to increase the window size simply by dragging the margins. On a Windows machine, you cannot increase the size of the window beyond the preset value by simply dragging the margins but (on Windows XP, 2000 and NT) you can change both the size of the screen buffer and the console window by right-clicking on the blue title bar of the MrBayes window and then selecting "Properties" in the menu that appears. Make sure the "Layout" tab is selected in the window that appears, and then set the Screen Buffer Size and Window Size to the desired values.

1.5 Getting Help

At the `MrBayes >` prompt, type `help` to see a list of the commands available in MrBayes. Most commands allow you to set values (options) for different parameters. If you type `help <command>`, where `<command>` is any of the listed commands, you will see the help information for that command as well as a description of the available options. For most commands, you will also see a list of the current settings at the end. Try, for instance, `help lset` or `help mcmc`. The `lset` settings table looks like this:

```
Parameter    Options                                  Current Setting
-------------------------------------------------------------------
Nucmodel     4by4/Doublet/Codon/Aa                    4by4
Nst          1/2/6                                    1
Code         Universal/Vertmt/Mycoplasma/
             Yeast/Ciliates/Metmt                     Universal
Ploidy       Haploid/Diploid                          Diploid
Rates        Equal/Gamma/Propinv/Invgamma/Adgamma     Equal
Ngammacat    <number>                                 4
Nbetacat     <number>                                 5
Omegavar     Equal/Ny98/M3                            Equal
Covarion     No/Yes                                   No
Coding       All/Variable/Noabsencesites/
             Nopresencesites                          All
Parsmodel    No/Yes                                   No
-------------------------------------------------------------------
```

Note that MrBayes 3 supports abbreviation of commands and options, so in many cases it is sufficient to type the first few letters of a command or option instead of the full name.

A complete list of commands and options is given in the command reference, which can be downloaded from the program web site (`www.mrbayes.net`). You can also produce an ASCII text version of the command reference at any time by giving the command `manual` to MrBayes. Finally, you can get in touch with other MrBayes users and developers through the mrbayes-users email list (subscription information at `www.mrbayes.net`).

1.6 Reporting and Fixing Bugs

If you find a bug in MrBayes, we are very grateful if you tell us about it using the bug reporting functions of SourceForge, as explained on the MrBayes web site (`www.mrbayes.net`). When you submit a bug report, make sure that you upload a data file with the data set and sequence of commands that produced the error. If the bug occurs during a MCMC analysis (after issuing the "mcmc" command), you can help us greatly by making sure the bug can be reproduced reliably using a fixed `seed` and `swapseed` for the `mcmc` command, and ideally also with a small data set. The Tracker software at SourceForge will make sure that you get email notification when the bug has been fixed in the source code on the MrBayes SVN repository at SourceForge. Note, however, that there may be some time before new executables containing the bug fix will be released.

Advanced users may be interested in fixing bugs themselves in the source code. Refer to section 7 of this manual for information on how to contribute bug fixes, improved algorithms or expanded functionality to other users of MrBayes.

1.7 License and Warranty

MrBayes is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 3 of the License, or (at your option) any later version.

The program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details (http://www.gnu.org/copyleft/gpl.html).

2 Tutorial: A Simple Analysis

This section walks you through a simple MrBayes example analysis to get you started. The following section gives you several additional tutorials, covering more advanced features of the program. The simple tutorial in this section is based on the `primates.nex` data file. It will guide you through a basic Bayesian MCMC analysis of phylogeny, explaining the most important features of the program. There are two versions of the tutorial. You will first find a Quick-Start version for

impatient users who want to get an analysis started immediately. The rest of the section contains a much more detailed description of the same analysis.

2.1 Quick Start Version

There are four steps to a typical Bayesian phylogenetic analysis using MrBayes:

1. Read the Nexus data file
2. Set the evolutionary model
3. Run the analysis
4. Summarize the samples

In more detail, each of these steps is performed as described in the following paragraphs:

1. At the `MrBayes >` prompt, type **execute primates.nex**. This will bring the data into the program. When you only give the data file name (primates.nex), MrBayes assumes that the file is in the current directory. If this is not the case, you have to use the full or relative path to your data file, for example `execute ../taxa/primates.nex`. If you are running your own data file for this tutorial, beware that it may contain some MrBayes commands that can change the behavior of the program; delete those commands or put them in square brackets to follow this tutorial.

2. At the `MrBayes >` prompt, type **lset nst=6 rates=invgamma**. This sets the evolutionary model to the GTR substitution model with gamma-distributed rate variation across sites and a proportion of invariable sites. If your data are not DNA or RNA, if you want to invoke a different model, or if you want to use non-default priors, refer to the rest of this manual and the Appendix for more help.

3.1. At the `MrBayes >` prompt, type **mcmc ngen=20000 samplefreq=100 printfreq=100 diagnfreq=1000**. This will ensure that you get at least 200 samples from the posterior probability distribution, and that diagnostics are calculated every 1,000 generations. For larger data sets you probably want to run the analysis longer and sample less frequently. The default sample and print frequency is 500, the default diagnostic frequency is 5,000, and the default run length is 1,000,000.

`--enable-fastlog` . With this option a fast approximation algorithm is used instead of the normal log math functions. Since this approximation algorithm can actually slow down the program on some computer architectures, this option is turned off by default.

`--enable-debug` . If you want to compile a debug version of MrBayes, you can use this option. This adds the appropriate flag for the GNU gdb debugger.

Before running the configure script, one can set a number of shell-environment variables to influence the compiler and it's flags.

`CC` . This variable defines which compiler to use. For example, `gcc` for the GNU compiler or `icc` for the Intel C compiler. The default setting is the GNU compiler.

`CFLAGS` . Sets the optimization flags for the compiler. The default is set to -O3, which yields good results for every platform. It is, however, possible to perform some tuning with this variable. We give a few possibilities below for some common processor types, assuming you are using gcc version 3. See the gcc manual for further information on optimization flags.

`Intel x86/AMD/AMD64`

Some compiler flags for gcc under unix and for gcc/cygwin under windows: `-march=X` , with X one of `pentium4` , `athlon-xp` or `opteron` . If you have one of these processors this will generate instructions specifically tailored for that processor.

`-mfpmath=sse` attempts to use the SSE extension for numerical calculations. This flag is only effective in combination with the above mentioned `-march` flag. This flag can provide a big performance gain. However, using this flag in combination with other optimization flags might yield numerically incorrect code. For example, one can set `-mfpmath=sse,386` , but this flag leads to incorrect results, when used in combination with `-march=pentium4` .

`-fomit-frame-pointer` saves some function overhead.

`-O2` instead of `-O3` turns on a smaller number of optimization flags. However, a number of optimizations turned on by `-O3` might give a slowdown instead of producing faster code, especially in combination with the `-mfpmath=sse` option.

`Mac G4 and G5`

Some compiler flags for gcc for OS X:

`-fast` . This flag is specific for the gcc version delivered by Apple. It turns on

a set of compiler flags which tries to optimize for maximum performance. This is the recommended setting if you have a G5 processor and this version of gcc. Code compiled with this flag will not run on a G4 processor.

`-O2` or `-O3` .

`-mcpu=X` , with `X` one of `G4` or `G5` .

Setting `-mcpu` or `-fast` on the Mac results in gcc enabling a number of different flags. Read the gcc manual carefully if you want to experiment with other flags. Compilation of MrBayes version 3.2 will look like,

```
~ $ tar zxf mrbayes-3.2.tar.gz
~ $ cd mrbayes-3.2
~/mrbayes-3.2 $ export CC=icc
~/mrbayes-3.2 $ export CFLAGS="-fast -w0"
~/mrbayes-3.2 $ ./configure
checking for gcc... icc
checking for C compiler default output file name... a.out
checking whether the C compiler works... yes
checking whether we are cross compiling... no
checking for suffix of executables...
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether icc accepts -g... yes
checking for icc option to accept ISO C89... none needed
checking how to run the C preprocessor... icc -E
checking for grep that handles long lines and -e... /bin/grep
checking for egrep... /bin/grep -E
checking for ANSI C header files... yes
checking for exp in -lm... yes
checking for a BSD-compatible install... /usr/bin/install -c
checking for readline in -lreadline... no
checking for 32 or 64 bits cpu... 32
configure: creating ./config.status
config.status: creating Makefile
config.status: creating config.h
icc -fast -w0 -DUSECONFIG_H   -c -o bayes.o bayes.c
icc -fast -w0 -DUSECONFIG_H   -c -o command.o command.c
icc -fast -w0 -DUSECONFIG_H   -c -o mbmath.o mbmath.c
icc -fast -w0 -DUSECONFIG_H   -c -o mcmc.o mcmc.c
icc -fast -w0 -DUSECONFIG_H   -c -o model.o model.c
icc -fast -w0 -DUSECONFIG_H   -c -o plot.o plot.c
icc -fast -w0 -DUSECONFIG_H   -c -o sump.o sump.c
```

```
icc -fast -w0 -DUSECONFIG_H   -c -o sumt.o sumt.c
icc -fast -w0 -DUSECONFIG_H   -c -o tree.o tree.c
icc -fast -w0 -DUSECONFIG_H   -c -o utils.o utils.c
icc -fast -w0 -DUSECONFIG_H  -lm  mb.c bayes.o command.o mbmath.o mcmc.o model.o plot.o s
IPO: performing multi-file optimizations
....
~/mrbayes-3.2 $
```

7.1.2 Compiling with Code Warrior or Visual Studio

We provide MrBayes project files for both Metrowerks Code Warrior and Microsoft Visual Studio in the source code package. All the relevant flags are set in these files, so you should be able to compile the code without any further modifications.

7.2 Compiling and Running the Parallel Version of MrBayes

Metropolis coupling or heating is well suited for parallelization. MrBayes 3 takes advantage of this and uses MPI to distribute heated and cold chains among available processors (Altekar et al., 2004). The optimal number of processors is therefore equal to the total number of chains (the number of runs times the number of chains per run, the default is 2 times 4 equals 8).

As of version 3.2, POOCH is no longer supported by the MrBayes team. If you wish to use POOCH for your Macintosh cluster and need support by the MrBayes team, you can buy a recent version of the POOCH application and donate it to MrBayes development team.

The MPI version for Unix clusters, including Xserve clusters, has to be compiled before you can run it. To tell the compiler that you want the MPI version, you need to run the configure script with the `--enable-mpi` option. The script and the resulting `Makefile` assume that you have the mpi compiler mpicc in your path and that your system is set up correctly so it can find the relevant libraries.

A typical make session would look as follows:

```
~/mrbayes-3.2 $ ./configure --enable-mpi
...
checking for mpicc... yes
...
```

```
~/mrbayes-3.2 $ make
mpicc -O3 -ffast-math -Wall -DUSECONFIG_H   -c -o bayes.o bayes.c
mpicc -O3 -ffast-math -Wall -DUSECONFIG_H   -c -o command.o command.c
mpicc -O3 -ffast-math -Wall -DUSECONFIG_H   -c -o mbmath.o mbmath.c
mpicc -O3 -ffast-math -Wall -DUSECONFIG_H   -c -o mcmc.o mcmc.c
mpicc -O3 -ffast-math -Wall -DUSECONFIG_H   -c -o model.o model.c
mpicc -O3 -ffast-math -Wall -DUSECONFIG_H   -c -o plot.o plot.c
mpicc -O3 -ffast-math -Wall -DUSECONFIG_H   -c -o sump.o sump.c
mpicc -O3 -ffast-math -Wall -DUSECONFIG_H   -c -o sumt.o sumt.c
mpicc -O3 -ffast-math -Wall -DUSECONFIG_H   -c -o tree.o tree.c
mpicc -O3 -ffast-math -Wall -DUSECONFIG_H   -c -o utils.o utils.c
mpicc -O3 -ffast-math -Wall -DUSECONFIG_H  -lreadline -lm  mb.c bayes.o command.o mbmath
```

This produces an MPI-enabled version of MrBayes called `mb` . Make sure that the mpicc compiler is invoked. It is perfectly normal if the build process stops for a few minutes on the `mcmc.c` file; this is the largest source file and it takes the compiler some time to optimize the code. How you run the resulting executable depends on the MPI implementation on your cluster. A simple approach would use LAM/MPI. First, the LAM virtual machine is set up as usual with `lamboot` . Then the parallel MrBayes job is started with a line such as

```
$mpirun -np 4 mb batch.nex > log.txt &
```

to have MrBayes process the file `batch.nex` and run all analyses on four processors (`-np 4`), saving screen output to the file `log.txt` . If you keep both a serial and a parallel version of MrBayes on your system, make sure you are using the parallel version with your `mpirun` command.

If your analysis takes a lot of time, we advise you to turn on checkpointing by setting the `checkpoint` option (`mcmcp checkpoint=yes`). In case of a crash or power failure, this allows you to restart the MrBayes application from the last checkpoint.

7.3 Working with the Source Code

MrBayes 3 is written entirely in ANSI C. If you are interested in investigating or working with the source code, you can download the latest (bleeding edge) version from the MrBayes SVN repository at SourceForge. You can access the SVN repository from the MrBayes home page at [http://sourceforge.net/projects/mrbayes SourceForge (http://sourceforge.net/projects/mrbayes)]. SourceForge gives de-

tailed instructions for anonymous access to the SVN repository on their documentation pages.

If you are interested in contributing code with bug fixes, the best way is to send a diff with respect to one of the recent revisions from the SVN repository to Maxim Teslenko (Maxim.Teslenko[at]nrm.se). Please, include in the bug report the SVN revision number of the MrBayes you are refaring to. We will include your fixes in the main development branch as soon as possible. If you would like to add functionality to MrBayes or improve some of the algorithms, please contact Maxim for directions before you start any extensive work on your project to make sure your additions will be compatible with other ongoing development activities. You should also consider whether you want to work with version 3 or version 4 of the program. We are currently shifting our focus to the development of MrBayes 4. Unlike version 3, which is written in C, this version will be written in C++ and our goal is to provide a cleaner, faster, and more extensively documented implementation of Bayesian MCMC phylogenetic analysis. This means, among other things, that the code will be better organized, and all important sections will be documented using [http://www.doxygen.org Doxygen] for easy access to other developers. You are welcome to examine this project as it develops by downloading the source code, doxygen documentation, or programming style directives from the MrBayes SVN repository at SourceForge.

7.4 Advanced Options

7.4.1 LSet UseGibbs Option

As described in the Gamma-distributed rates section, MrBayes can accommodate rate heterogeneity across sites using a discrete approximation to the Gamma-distribution. The discrete approximation to the Gamma-distribution is a form of hidden Markov model, in which there are `ngammacat` rate categories that any site can belong to. There is a "hidden" state that identifies the appropriate rate category for each site. Because we cannot see the rate category for a site, we must treat it as an unknown variable. The `LSet UseGibbs` option controls how this variable is handled.

When `LSet UseGibbs=no` is in effect, the likelihoods calculated by MrBayes will be comparable to the likelihood from other software (and versions of MrBayes

earlier than 3.2). In this mode, a weighted sum over all rate categories is performed to calculate each site's likelihood. The weights are the prior probabilities that a site would belong to that category. Because the gamma is discretized by breaking it up into `ngammacat` equal size chunks, the probability that any site i, would be in rate category c is simply `1/ngammacat` (and this quantity is independent of the site of the rate). The likelihood for site i is a sum over all categories c of the quantity

(1/ngammacat)L(i $|c$)

where L(i $|c$) is the likelihood of site i conditional upon it being in category c. Typically (at least for datasets with a large number of taxa), one of the rate categories contributes the vast majority of the likelihood to this sum, because its conditional likelihood is much larger than the conditional likelihoods from the other rate categories. Thus the likelihoods under this model are close to `1/ngammacat` times the conditional likelihood of the best fitting rate category for each site.

The fact that the site likelihoods are dominated by one term implies that it is wasteful to calculate the conditional likelihoods over all `ngammacat` rate categories –most of the calculations will be lost in the rounding error of the final summation over rate categories.

The `LSet UseGibbs=yes` option addresses this concern. Instead of calculating a site's likelihood by summing over all of the hidden states (the rate categories), an indicator variable that corresponds to the rate category for a site is sampled during the MCMC. The "likelihood" reported for each site is actually the conditional likelihood - the likelihood of the site conditional upon it belonging to rate category. Each site has its own value for the hidden state. The fact that we are uncertain of each site's "true" rate category is accommodated by sampling with MCMC instead of summation.

Conveniently, when you calculate the likelihood by summing over all rate categories, then you are also in position to calculate the posterior probability that each site belongs to a particular category. To update the hidden states for each site, you have to:

1. Calculate the conditional likelihood for each rate category.

2. Calculate the full site-likelihood by summing over all rate categories (weighted by their prior probabilities).

3. Calculate the posterior probability of each of the hidden state assignments, by dividing the conditional likelihood for each rate category by the site likelihood (so that the probabilities sum to one).

4. Select a new hidden state for each site by choosing a randomly – giving each rate category a probability of being chosen that is equal to the posterior probability that the site belongs to that category

This is a Gibbs_sampling Gibbs Sampling update of the hidden state. The `LSet Gibbsfreq` option controls how frequently the hidden states are updated. Thus the likelihood for all `ngammacat` categories only has to be calculated in the iteration in which the hidden state is being updated In all other iterations the likelihood is calculated from only the rate category to which that site is currently assigned.

Note that in the `LSet UseGibbs=no` mode, the priors associated with each rate category (the `1/ngammacat` terms) are used in calculating the site likelihood. In the `LSet UseGibbs=yes` mode these priors are dealt with when rate categories are sampled. Thus, when using Gibbs Sampling the likelihood reported is not decreased by weighting it by a prior probability (thus the likelihoods that are reported will be higher).

8 Acknowledgements

We would like to acknowledge the invaluable help we have received from students, colleagues and numerous users of MrBayes; they are too many to name them all here. Often, we have been overwhelmed by the generosity with which people have shared ideas, bug fixes and other valuable tips with us. This feedback alone makes all the hours we have put into developing MrBayes worthwhile. Thank you, all of you!